# Chapter 1

# Introduction

This monograph presents a framework for learning in a distributed data scenario with decentralized decision making. We have based our framework in Multi-Agent Systems (MAS) in order to have decentralized decision making, and in Case-Based Reasoning (CBR), since the lazy learning nature of CBR is suitable for dynamic multi-agent systems. Moreover, we are interested in autonomous agents that collaboratively work as ensembles. An ensemble of agents solves problems in the following way: each individual agent solves the problem at hand individually and makes its individual prediction, then all those predictions are aggregated to form a global prediction. Therefore, in this work we are interested in multi-agent learning techniques to improve both individual and ensemble performance. In this chapter we are going to motivate our framework and state our research goals. Then, we will present a road-map of the contents of this thesis.

## 1.1 Motivation

Data of interest is distributed among several sources in many real world applications. Therefore, traditional machine learning techniques cannot be directly used, since they assume a centralized access and control of the data. In order to deal with the distributed nature of data of interest, several approaches have been proposed.

A first approach is to collect data from the different data sources, and store it in a centralized repository, where machine learning techniques can be used. However, in many domains this approach is not desirable or even not feasible for a variety of reasons. For instance for property rights, bandwidth limitations, or because of management concerns (since data owners may not be wiling to cede their data to a centralized repository because they want to maintain control over their data).

A second approach is based on the fact that many machine learning techniques can be decentralized. For instance, certain decision trees techniques can

be used in a distributed way [Caragea et al., 2003] by locally computing certain statistics at each data source, and then sending those statistics to a central repository where a decision tree can be learnt. However, this second approach only solves the problem of bandwidth limitation and is only applicable to machine learning techniques that can be decentralized.

The previous two approaches correspond respectively to *data warehousing* and *distributed machine learning*. Moreover, they both share two assumptions: a) that the *only* problem is that data is distributed, and b) that a single model of all the data is going to be constructed. Let us develop both issues in more detail.

In many applications the fact that data is distributed among several sources is not the only problem. The problem is that the different data sources may correspond to different partners or organizations, and that those organizations may consider their cases as assets and may not be willing to allow other organizations to have access to their data either because of ownership rights or management concerns. However, these organizations would be interested in benefiting from the collaboration with other organizations but keeping the control of their data.

A way to deal with the privacy rights and management concerns may be Multi-Agent Systems (MAS) [Durfee and Rosenschein, 1994, Jennings, 1993, Woolridge, 1992], a sub-field of distributed artificial intelligence that studies how autonomous entities (a.k.a. *agents*) interact, in a collaborative or competitive way. Researchers in multi-agent systems focus mainly on architectures that can support agent systems [Esteva et al., 2001], and on distributed mechanisms to coordinate multiple agents so that they can jointly accomplish a given task [Jennings, 1993]. The intersection of learning and multi-agent systems is called *Multi-Agent Learning* (MAL) [Stone and Veloso, 2000], and addresses the integration of learning in multi-agent systems. However, it is a relatively new field and a lot of work still remains to be done. Moreover, most of the work focuses on reinforcement learning and evolutionary algorithms.

We have to take into account the difference between a distributed algorithm and a multi-agent system: in a distributed algorithm there is a global *goal* (and there are several processes running in parallel to accomplish that goal), while in a multi-agent system each individual agent has its own goals. The joint goals emerge from the interaction among several agents following an interaction protocol: eventually a group of agents may collaborate together to solve a task, but only if that task is beneficial for each one's goals. Thus, multi-agent systems are a suitable tool to preserve the privacy and management concerns in the distributed data scenario, where each organization can be modelled as an agent that has control over its private data. Moreover, two organizations will only collaborate when they both are interested in collaboration.

Concerning the issue of building a single model, it is not obvious that building a single model of the data is always the best solution. For instance, ensemble learning is a subfield of machine learning based on constructing several models of the same data and then combine them in order to reduce error with respect to using a single model. Thus, at least in principle, having multiple models of data

is better than having a single model. Ensemble learning methods are centralized and, given a training set, construct a set of different classifiers by training each classifier with a variation of the training set or with a different learning method. Ensemble methods reduce error with respect to using a single model for three main reasons [Dietterich, 2000]: first, they enhance the expressive power of the classifiers (since the ensemble can express hypothesis that cannot be expressed with a single classifier); second, they reduce the impact of having a small training sample (since a small sample increases the likelihood of finding a wrong hypothesis, and the aggregation of several hypotheses is more likely to perform better); and third, they reduce the problem of getting stuck in a local minimum during learning (since each classifier is expected to find a different local minimum, and their aggregation is expected to perform better). Moreover, we will call the classification error reduction achieved by ensemble methods the *ensemble effect*.

A basic assumption of ensemble learning methods is a centralized control over the data. This assumption does not hold in multi-agent systems where control is decentralized, since each individual agent controls part of the data and each agent is autonomous. Therefore, ensemble learning techniques cannot be directly applied to build multiple models in a distributed data setting modelled using multi-agent systems. Another problem is that ensembles must satisfy some preconditions in order to perform well (that we will refer to as the "preconditions of the ensemble effect"), and in an open multi-agent system we have no guarantee that the individual agents satisfy those preconditions. Thus, if the benefits of the ensemble effect are desired, alternative techniques are needed.

In this work, we are going to present a framework to deal with learning in distributed data, based on multi-agent systems, and where we are interested in using multiple models of the data. Moreover, due to the open and dynamic nature of multi-agent systems, we are interested in Lazy Learning techniques, and specially Case-Based Reasoning (CBR) [Aamodt and Plaza, 1994]. Lazy learning techniques are better suited for open and dynamic systems than eager learning techniques, since they are not sensitive to changes in the data, while eager learning techniques have to rebuild (or adapt) their models of the data every time that data changes.

Case-Based Reasoning (CBR) is a specific type of lazy learning, that consists of storing problem solving experiences (called *cases*) so that they can be reused to solve future problems. CBR basically relies on the assumption that similar problems require similar solutions. A typical CBR system solves a problem by retrieving cases stored in its case memory (called the *case base*) that are similar to the problem at hand, and reusing the solution of the retrieved cases to solve the problem. Once the proposed solution for the new problem has been revised, a new case is created and it can be retained in the case base. This problem solving cycle is known as the R4 model [Aamodt and Plaza, 1994], that divides the activity of a CBR system in four processes: *retrieve*, *reuse*, *revise*, and *retain*.

Classical CBR considers a single case base with which to solve problems. Applying CBR to multi-agent systems arises several issues. For instance, the reuse process in the R4 model assumes that cases from a single case base have

been retrieved. However, in a multi-agent system several agents may control different case bases. Moreover, agents may consider their case bases private, and thus the problem is not simply to retrieve cases from several case bases, but how several agents (each one controlling its case base) can collaborate to solve problems using CBR, without violating neither autonomy of agents nor the privacy of data. Moreover, case retention is not obvious either: in a classical CBR system a case is retained into the case base after being solved. However, in a multi-agent system, where a group of agents may have collaborated to solve a case, it is not clear which agent or agents should retain that case. Therefore, at least two new issues appear: how to solve problems in a collaborative way, and how to perform retention in a collaborative way.

## 1.2 The Framework

In this thesis we will present the *Multi-Agent Case Based Reasoning Systems* ($\mathcal{M}$AC) framework for learning in distributed data settings with decentralized decision making. Agents in a multi-agent system (MAS) have autonomy of decision, and thus control in a MAS is decentralized. Moreover, $\mathcal{M}$AC systems take a social agents approach based on electronic institutions [Esteva et al., 2001]. In electronic institutions, coordination among agents is performed by means of shared interaction protocols. Basically, an interaction protocol defines a set of interaction states, and the set of actions that each agent can perform in each interaction state. Each agent uses individual decision policies to choose form the set of possible actions at each interaction state. In the $\mathcal{M}$AC framework, agents collaborate by means of *collaboration strategies*, consisting on an interaction protocol and a set of individual decision policies. Thus, the electronic institutions offers us a framework where autonomy in decentralized decision making is preserved.

Each individual agent in a $\mathcal{M}$AC system is capable of individually learn and solve problems using CBR, with an individual case base. Moreover, each case base is owned and managed by an individual agent, and any information is disclosed or shared only if the agent decides so. Thus, this framework preserves the privacy of data, and the autonomy to disclose data. Therefore, the $\mathcal{M}$AC framework extends the case-based reasoning paradigm to multi-agent systems. Moreover, notice that since each individual agent is an individual case based reasoner, agents have the ability to learn individually.

The focus of this thesis is investigating ensembles of agents. Specifically, we are interested in studying how to organize agents into ensembles, and how they can collaborate to achieve the ensemble effect. For this purpose, we need to address other issues such as determining when an agent should solve a problem individually or organizing an ensemble, and determining which agents should be present in an ensemble. Moreover, we are also interested in studying how individual and ensemble performance can be improved. For this purpose, we need to address several other issues such as learning how to select the members of an ensemble, learning how to improve individual performance maintaining (or

even improving) ensemble performance, determining how to redistribute cases among the agents to achieve better distributions (in terms of performance), and deciding which agents should retain new cases so that individual and ensemble performance improves.

In order to address those issues we will design collaboration strategies, i.e. we will design interaction protocols and individual decision policies. Thus, these collaboration strategies will allow agents to form ensembles and to improve their performance as a result of individual decisions made in a decentralized way. In the next section we will present a detailed list of our research goals in the $\mathcal{M}$AC framework.

## 1.3 The Goals

The main goal of the thesis is to study the effects of distributed data and decentralized individual decision making in learning processes, and specifically in a multi-agent setting where individual agents own different parts of the data.

Moreover, in this thesis we have also several goals related with CBR, multi-agent systems and ensemble learning:

- The first goal is the integration of the three related areas (ensemble learning, case-based reasoning and multi-agent systems) and formally define the Multi-Agent Case Based Reasoning framework ($\mathcal{M}$AC).
- How to achieve the *ensemble effect* in multi-agent systems by forming *committees of agents*. Thus, allowing agents to improve their performance as an ensemble as a result of their individually made decisions.
- Analyze the ensemble effect and its preconditions in a wide range of situations so that measures can be defined to characterize ensembles, and thus predicted their performance. This measures are required so that agents trying to behave as an ensemble can measure how well will they perform as an ensemble and decide which actions should be taken to improve their ensemble performance.
- Develop learning techniques to improve the performance of the agents (both individually and as an ensemble). Specifically, we are interested in two types of leaning: learning processes that allow agents to improve individual problem solving performance, and learning processes that allow agents to improve their collaboration, i.e. learning when to collaborate and with whom to collaborate.
- Extend the Case-Based Reasoning paradigm to deal with multi-agent systems, in such a way that agent autonomy, data privacy, and individual data control are preserved in the autonomous agents. The four processes of CBR (retrieve, reuse, revise and retain) have to be rethought. Specifically, in this thesis we focus on how reuse and retain can be adapted to work in multi-agent systems.

- Develop techniques to perform the reuse process of CBR in a decentralized way. Decentralized reuse would be preferable to decentralized retrieve under certain conditions, since decentralized reuse can preserve privacy of the data while decentralized retrieve cannot. Decentralized reuse should be able to determine a global prediction through a collaborative process (based for instance in voting or in any other aggregation mechanism) among several agents that have performed the retrieval process individually.

- Develop techniques to perform the retain process of CBR in a decentralized way. Decentralized retain raises several new problems with respect to classical retain involving a single case base. For instance, using decentralized retain, a group of agents solving a problem has to decide not only if a case is going to be retained, but which agent or agents will retain it. Moreover, decentralized retain has to take into account the performance of the agents when they act as an ensemble, in addition to the performance of the agents solving problems individually.

- Develop techniques for decentralized data redistribution. Since in a multiagent system we cannot make any assumption about the initial distribution of data among the agents, it would be interesting to study data redistribution techniques that rely in decentralized control and that preserve the autonomy of the agents. Redistribution techniques have the goal of achieving a distribution of data that improves both individual and ensemble performance.

## 1.4 The Thesis

In this section we will present a road map of the thesis, shortly summarizing the contents of the rest of the chapters and appendices. Figure 1.1 shows a condensed view of the contents of the thesis.

- **Chapter 2** presents an overview of the state of the art in the areas related to the research presented in this thesis. First, related work in Ensemble Learning is presented, emphasizing in the work related to the *ensemble effect* and on different methods for creating ensembles. Then, related work on Case-Based Reasoning is considered, specifically the work related to retain techniques and to explanations generation. Finally, recent work on multi-agent learning is reviewed. Specifically, four areas of multi-agent learning are reviewed: reinforcement learning and genetic algorithms in multi-agent systems (since these two are the most applied techniques to multi-agent learning) and also Case-Based Reasoning in multi-agent systems.

- **Chapter 3** presents the $\mathcal{M}$AC framework for ensemble case based learning. First, the relevant multi-agent systems concepts for our research are

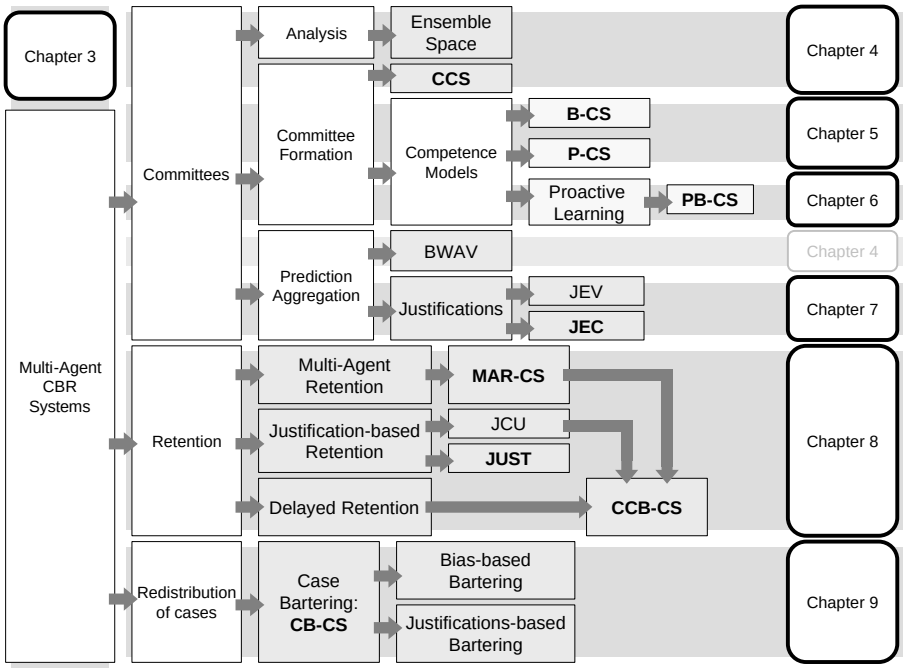


Figure 1.1: Graphical overview of the contents of this thesis.

introduced. Specifically, collaboration strategies are defined and a formalism to specify interaction protocols is presented. Then, *feature terms*, the formalism used to represent knowledge in our work, are presented jointly with the NOOS Agent Platform, a LISP based agent platform specifically designed to incorporate integrate learning and reasoning techniques using feature terms as representation language. Finally, the *Multi-agent Case Based Reasoning Systems* are formally defined and our approach to multi-agent learning is presented from a Case-Based Reasoning perspective.

- **Chapter 4** presents the concept of a *committee* (a group of agents that joins together to solve a problem using a voting system). A committee is the organizational form of an "ensemble of agents" from the point of view of multi-agent systems, defined to study the ensemble effect in multi-agent systems. Specifically, the *Committee Collaboration Strategy* with which a group of agents can act as a committee is presented, and the *Bounded Weighted Approval Voting* is introduced as a voting system specifically designed for committees of agents that use CBR to solve problems. Chapter 4 also presents the *ensemble space analysis*, an analytical tool to characterize committees of agents and that we will use in the rest of the thesis as a way to analyze the performance of a committee. Later, Chapters 5, 6, and 7 present extensions of the basic Committee Collaboration Strategy.

- **Chapter 5** presents the idea of the *dynamic committee* collaboration strategies, that are strategies that convene different committees of agents depending on the problem that has to be solved. Specifically, two different dynamic committee collaboration strategies, namely the *Peer Counsel Collaboration Strategy* and the *Bounded Counsel Collaboration Strategy*.

- **Chapter 6** deals with how agents can learn to collaborate better. For that purpose we introduce *competence models*, functions that assess the likelihood of the solution provided by an agent (or set of agents) to be correct. Next, we present the *proactive learning* of competence models as a way in which individual agents can learn when to collaborate and with whom to collaborate. Competence models can be autonomously learnt by the agents interacting with other agents. Finally, the *Proactive Bounded Counsel Collaboration Strategy* is presented, combining dynamic committees with proactive learning.

- **Chapter 7** introduces the notion of *justification*. A justification is the explanation given by a CBR agent (or any other problem solving system) of why it has considered the solution of a specific problem to be correct. In this chapter, we use justifications to deal with the issue that it cannot be taken for granted that the agents in a $\mathcal{M}$AC system satisfy the preconditions of the ensemble effect. For that purpose, we will show that justifications can be examined by some agents to assess the confidence of the predictions made by other agents. Then, we will show how to use this information to define an aggregation mechanism to determine which is the solution with highest confidence, namely, the *Justification Endorsed Voting System* (JEV). Finally, we present the *Justification Endorsed Committee Collaboration Strategy* that uses JEV to improve the performance of committees by weighting the individual votes according to the confidence assessed to their predictions.

- **Chapter 8** addresses the issue of case retention in Multi-Agent Case-Based Reasoning Systems. Specifically, three ideas are introduced: collaboration strategies for case retention, the assessment of the case utility using justifications, and delayed retention. First, several collaboration strategies for case retention are presented, showing that they can outperform individual retention strategies. Then, we introduce the *Justification-based Case Utility* (JCU), a case utility function based on justifications that can assess how useful will be a case for an agent. Moreover, we present the *Justification-based Selection of Training Examples*, a case base reduction technique that uses JCU to generate a reduced case base with the same problem solving accuracy as the original one. Thirdly, we show that delayed retention can improve performance with respect to retention strategies that consider cases one by one. Finally, the *Collaborative Case Bargaining Collaboration Strategy* is presented as a retention strategy that combines the three ideas presented in the chapter.

- **Chapter 9** presents a new family of collaboration strategies that use the idea of *case bartering*. Case bartering is designed as a way to deal with the problem of finding a redistribution of cases among the case bases of the agents so that they perform better both as individuals and as a committee. Specifically, we present two basic case bartering strategies: the *Bias Based Case Bartering Collaboration Strategy* and the *Justifications Based Case Bartering Collaboration Strategy*. The first one is inspired on the ensemble space analysis presented in Chapter 4, and is based on decreasing the bias in the individual case bases of the agents with the goal of boosting both the individual performance of agents and their ensemble performance. The second strategy is inspired in the case utility assessment based on justifications and allows each agent to obtain high utility cases with the goal of improving both their individual and ensemble performance.

- **Chapter 10** first summarizes the work presented in this thesis. Then, the contributions with respect to ensemble learning, case-based reasoning and multi-agent systems are presented. The chapter closes with a discussion of future lines of research.

- **Appendix A** presents a comprehensive list of the notation used in all the chapters of this thesis.

- **Appendix B** presents an overview of the NOOS agent platform, that we have used in our research.

- **Appendix C** presents a probability assessment technique used in the Proactive Bounded Counsel Collaboration Strategy (Chapter 5), and in the Justification-based Selection of Training Examples (Chapter 8). This technique determines a confidence interval for classification accuracy estimations.

## 1.5 Notation

In the remainder of this thesis we have followed the following notation conventions:

- $A_i$: is used for agents (different subindexes denote different agents).

- $c_i$: is used for cases.

- $\mathbf{R}$: boldface upper case letters are used for tuples (or records).

- $\langle f_1, ..., f_n \rangle$: angle-bracketed lists are also used for tuples (or records).

- $\mathbf{R}.f_i$: dot notation is used to refer to the value of the field $f_i$ of the tuple $\mathbf{R}$.

- $C$: when elements of a certain kind are noted with a lower case letter, sets of such elements are noted with an upper case letter. For instance, since cases are noted with letter $c$, sets of cases — a.k.a. case bases — are noted with letter $C$.
- $\mathcal{A}$: when elements of a certain kind are noted with an upper case letter, sets of such elements are noted with a calligraphic upper case letter. For instance, since agents are noted with an upper case letter $A$, sets of agents are noted with a calligraphic letter $\mathcal{A}$.
- $\mathbb{A}$: when elements of a certain kind are noted with a calligraphic letter, sets of such elements are noted with a "blackboard bold" letter. Moreover, since elements noted with a calligraphic letter are usually sets, elements noted with blackboard bold letters are usually called "collections of sets" (for not using "sets of sets" that would be confusing).
- $\#(\mathcal{A})$: denotes the cardinality of the set $\mathcal{A}$.

Appendix A contains a comprehensive list of all the notation used throughout this thesis.